



[hash] Bảng băm thăm dò - Hàm Insert - 1 (gần giống với đề thi cũ)



[Download the code template](#)

ÔN TẬP LÝ THUYẾT: đây chỉ là 1 mã giả tham khảo. Điều kiện dừng của quá trình băm lại có thể khác tùy theo yêu cầu của GV. Nếu không có yêu cầu gì đặc biệt thì thông thường, điều kiện dừng mặc định sẽ là $i=m$ (như trong slide).

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác thêm phần tử:

Đầu vào:

- Bảng băm T có kích thước m,
- Hàm băm lại $h_i(k)$.
- Phần tử có khóa k cần thêm

Đầu ra:

- Bảng băm T.
 - Vị trí của phần tử được chèn hoặc -1 nếu gặp lỗi.
- Ký hiệu DELETED và NIL được dùng để đánh dấu vị trí đã được xóa và vị trí trống

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, KHU PHỐ 6, PHƯỜNG LINH TRUNG, QUẬN THỦ ĐỨC, TP. HỒ CHÍ MINH

[T] 08 3725 2002 101 | [F] 08 3725 2148 | [W] www.uit.edu.vn | [E] info@uit.edu.vn



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác thêm phần tử:

HASH-INSERT(T, k)

```

1.   i = 0
2.   do
3.       j = h(k, i)
4.       if T[j] == NIL or T[j] == DELETED
5.           T[j] = k
6.           return j
7.       else i = i + 1
8.   while i < m
9.   return -1

```

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, KHU PHỐ 6, PHƯỜNG LINH TRUNG, QUẬN THỦ ĐỨC, TP. HỒ CHÍ MINH

[T] 08 3725 2002 101 | [F] 08 3725 2148 | [W] www.uit.edu.vn | [E] info@uit.edu.vn

Hãy hoàn thành chương trình cho phép nhập 1 danh sách học sinh và lưu vào bảng băm.

Cho biết:

- Hashtable là loại bảng băm xử lý đụng độ bằng phương pháp thăm dò (còn gọi là địa chỉ mở).
- Hàm băm lại sử dụng phương pháp thăm dò tuyến tính: $h(\text{key}, i) = ((\text{key} \% M) + i) \% M$. Với M là kích thước bảng băm

Hệ số tải của bảng băm được định nghĩa là 0.7 , nghĩa là bảng băm nhận thêm các bộ phận sẽ được lưu trong bảng băm không quá 70% kích thước của bảng băm.

- Hằng ký hiệu EMPTY 0 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm đang còn trống.

- Hằng ký hiệu DELETE -1 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm có phần tử trước đó nhưng đã bị xóa.

- Số lượng phần tử cần thêm vào bảng băm không quá 1000. Kích thước bảng băm không quá 1000 phần tử.

- Hàm void CreateHashtable(Hashtable &, int) để khởi tạo bảng băm, tức cấp phát vùng nhớ, gán giá trị ban đầu cho phần tử mảng và các thuộc tính

- Hàm Insert(Hashtable &, Hocsinh) để thêm một phần tử có kiểu dữ liệu Hocsinh vào một bảng băm kiểu Hashtable. Hàm Insert sẽ trả về kết quả 1 nếu thêm phần tử thành công và trả về 0 nếu không thêm phần tử được.

- Hàm PrintHashtable(Hashtable) để xuất bảng băm. Định dạng xuất như mô tả trong OUTPUT

- Hàm DeleteHashtable(Hashtable &) để xóa bảng băm, bằng cách giải phóng vùng nhớ động, reset các thuộc tính còn lại của bảng băm.

INPUT:

- Một số nguyên dương $m < 1000$, là kích thước của bảng băm

- Một số nguyên dương $n < 1000$, là số lượng học sinh sẽ thêm vào bảng băm

- Thông tin của n học sinh. Thông tin của 1 học sinh được nhập theo thứ tự {mã số, họ tên, năm sinh, giới tính, TBK}

OUTPUT:

- Xuất toàn bộ bảng băm, theo ví dụ và template.

+ Tại địa chỉ nào có lưu học sinh, xuất thông tin của học sinh. VD: [20520069, Vu The Hoai Son , 1, 2003, 7.7]

+ Tại địa chỉ nào rỗng thì xuất [0, , , ,]

SV copy đoạn code bên dưới để xuất đúng format.



```

cout << "[" << hs.Maso << ", " << hs.HoTen << ", " << hs.Giay << ", " <<
hs.Namsinh << ", " << hs.TBK << "]"\\n";
else
    cout << "[" << hs.Maso << ", " << " , " << " , " << " , " << "]"\\n";

```

VÍ DỤ:

Input	Output
101	
100	
20520001	
Nguyen The Nhu Quynh	
2003	
0	[20520069, Vu The Hoai Son , 1, 2003, 7.7]
5.3	[20520070, Duong Thanh Tam , 0, 2003, 8.1]
20520003	[20520071, Ho Quoc Tai , 1, 2002, 6]
Le Hoang Quan	[0, , , ,]
2003	[20520073, Huynh Le Tan Tai , 1, 2002, 7.6]
1	[20520074, Huynh Huu Tam , 1, 2004, 5.6]
9.7	[0, , , ,]
....
....	
...	

13. [Inclass#11] Hash Table (Pre-Intermediate).mở lại .không tính điểm

14 problems with a total score of 1400

#	Problem	Score
1	[hash] Loại bỏ số trùng (dùng kỹ thuật băm dạng đơn giản nhất, ko có xung đột).	100



2	[hash] Tìm giá trị chỉ xuất hiện 1 lần trong ma trận (dùng kỹ thuật băm dạng đơn giản nhất,ko có xung đột).	100
3	[hash] Ký tự còn thiếu (dùng kỹ thuật băm dạng đơn giản nhất,ko có xung đột).	100
4	[hash] Find MEX (dùng STL.map).	100
5	[hash] Find MEX (dùng STL.set).	100
6	[hash] Find MEX (dùng STL.vector).	100
7	[hash] Kiểm kê (dùng STL).	100
8	[hash] Login 1 (dùng STL).	100
9	[hash] Create a hash table - dò tuyến tính (using array).	100
10	[hash] Create a hash table - dò bậc 2 (using array).	100
11	[hash] Bảng băm nối kết - Hàm Insert - 1 (gần giống với đề thi cũ).	100
12	[hash] Bảng băm nối kết - Hàm Search - 1 (gần giống với đề thi cũ).	100
13	[hash] Bảng băm thăm dò - Hàm Insert - 1 (gần giống với đề thi cũ).	100
14	[hash] Bảng băm thăm dò - Hàm Search - 1 (gần giống với đề thi cũ).	100



Submit

upload source code

Chọn tệp

Không có tệp nào được chọn

Select language

C++ (0.5s, 50MB)

Submit

[Code editor](#)